

---

# Deep learning for prediction of amyotrophic lateral sclerosis using stacked auto encoders

---

Sindhu P. Menon

Jain College of Engineering and Technology,  
Hubli, India  
Email: sindhu33in@gmail.com

**Abstract:** Healthcare is an emerging area in big data. Raw data contains lot of noise in it, hence cannot produce good results when processed. There is a need to improve the quality of data. This study shows how the prediction accuracy can be improved if the quality of data is improved. Previous work on issues related to variety and veracity have already been cited. Here the issues related to prediction are addressed. The dataset contains 1,047,253 records of patients having amyotrophic lateral sclerosis (ALS). Missing data values are filled and later used for prediction. Predicting the progression of the disease was calculated using stacked auto encoders. The results were compared with traditional techniques like random forest and support vector machine. A similar study was conducted using random forests and the accuracy obtained was only 66%. This paper presents a study on how to predict the progression of ALS using deep learning and an accuracy of 88% was achieved which is far more than the accuracy obtained on raw data. The study thus demonstrates the fact that accuracy increases with better data.

**Keywords:** prediction; amyotrophic lateral sclerosis; ALS; deep learning; auto encoders.

**Reference** to this paper should be made as follows: Menon, S.P. (2020) 'Deep learning for prediction of amyotrophic lateral sclerosis using stacked auto encoders', *Int. J. Big Data Management*, Vol. 1, No. 2, pp.119–134.

**Biographical notes:** Sindhu P. Menon completed her PhD in the area of Big Data from JNTU, Hyderabad. She has about 15 publications in peer reviewed journals and conferences. She is presently working as a faculty in the Department of Computer Science and Engineering at Jain College of Engineering and Technology, Hubballi. Her research interests are machine learning, deep learning, big data, and data mining

---

## 1 Introduction

Motor neuron disease (MND) is a generally uncommon however fatal neurodegenerative problem of the motor system in adults. MND is a staggering condition with obscure etiology and no present cure. The symptoms in MND are assorted and challenging. They incorporate weakness, spasticity, constraints in mobility and exercises of day by day living, correspondence shortfalls and dysphagia, and in those with bulbar contribution, respiratory bargain, weariness and rest issue, pain and psychosocial trouble. The current evaluation for the progression of the disease is based on ten features which take values

ranging from 0 to 4.4 indicates normal and 0 is complete disability in that activity. The various features are walking, breathing, cutting, turning, speech, salivation, swallowing, handwriting, cutting and dressing. A value of 0 in speech indicates that the person is able to speak normally whereas a value of 4 indicates his inability to talk. The aim of this study was to develop a model that can predict the progression of disease.

Bayesian networks, random forests (RF), k-means and decision trees have been experimented to determine the progression of the disease on a variety of datasets. The various tools available is explained by Menon and Hegde (2015). In Küffner et al. (2015) algorithms for the prediction of disease progression of 1,822 ALS patients from standardised, anonymised phase 2/3 clinical trials were performed. Three patients showing an ALSFRS slope of less than  $-1.1$  points/month were considered fast, seven patients with a slope greater than  $-0.5$  points/month were considered slow, and the remaining four patients were considered average.

Earlier work on this includes interpolating medical image data (Achiepo et al., 2015) using convolution based interpolation methods, applying Lagranges interpolation (Matsumura, 1992) to determine similarity measure for medical diagnostics and to reconstruct images, filling missing data using decision trees (Marwala, 2009) and so on. Kim et al. (2014) have predicted the progression of amyotrophic lateral sclerosis (ALS) and have achieved 66% accuracy. The same is accomplished using RF algorithm. The parameters used for prediction of the disease are forced vital capacity (FVC) and Height. In future work they have proposed that these experiments could be carried using other parameters and other algorithms to see if efficiency could be increased. Sindhu and Hegde (2017) have suggested ways of improving the quality of data. Later in 2017, Schuster et al. were able to achieve survival prediction accuracy of 66.67% with sensitivity of 62% and 70% specificity. Similar studies were conducted by Küffner et al. (2015) using k-means algorithm to predict the progression of the disease. They were able to achieve 83.2% accuracy on a small dataset. In 2017, Ong et al. conducted a similar study. They predicted the functional decline and survival in ALS using Time series models. They were able to get an accuracy of 82%. A similar study was done by Menon (2019) wherein they achieved higher accuracy. The objective of this work is to develop and test a method which can predict the progression of ALS with better accuracy. Thus, the motivation for the work related to prediction of progression of the disease.

Section 1 deals with the introduction. In Section 2, the dataset and architecture of the proposed system are discussed. Stepwise execution of the algorithm is also shown here. Section 3 deals with the results and discussion on the same. Finally, in Section 4 the conclusions of the work is there wherein an accuracy of 87% is obtained using deep learning technique.

## **2 Proposed system**

### *2.1 Dataset*

Data used in this article was obtained from the pooled resource open-access ALS clinical trials (PRO-ACT) database. In 2011, Prize4Life, in collaboration with the Northeast ALS consortium, and with funding from the ALS therapy alliance, formed the PRO-ACT consortium (PRO\_ACT Database).

The database consists of 11 datasets-ALSFRS, death report, demographics, family history, FVC, laboratory data, riluzole use, slow vital capacity, subject ALS history, treatment group, and vital signs (PRO\_ACT database). The major symptoms of ALS are muscle weakness, paralysis, drooling, gagging, muscle cramps, voluntary muscles and significant weight loss. This study we have used data from ALSFRS, demographics and laboratory. Demographics will contain information like height, weight, gender, age. lab details will include ALS value, values for various voluntary activities. Many of these values will be missing in the database. Hence, there is a need to fill in these values for accurate analysis.

## 2.2 System architecture

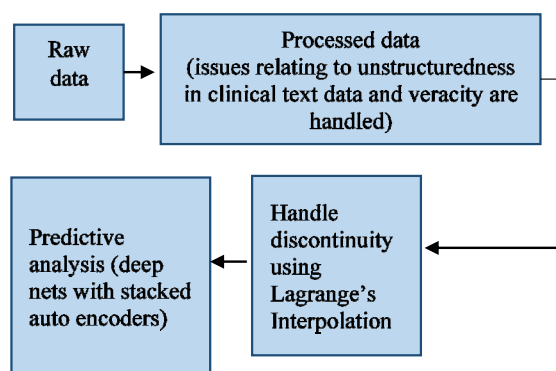
The objective behind developing this model is to build a system where the prediction accuracy increases if quality of data is improved.

The data obtained from the PRO-ACT database is the raw data. This is first converted to a format which HBASE can handle. In Ong et al. (2017) the issue related to unstructuredness in medical text data has been handled to some extent. The data is converted to a key-value format and then stored as explained in Ong et al. (2017).

Since the data contains missing values, these values have to be filled before further processing. The issues relating to veracity is handled in Sindhu and Hegde (2015) using modified moving average (MMA). Comparison with auto regression tree and moving average is also done in Sindhu and Hegde (2015) but it was found that MMA yields better results than the other two.

The output from Sindhu and Hegde (2015) is given as an input to the proposed work. The reason for using Lagrange's Interpolation to handle discontinuity is discussed in Meijering et al. (2001). In case there is a stoppage in the arrival of data, then the analysis should not stop. In such a situation, data is substituted for the missing values using Lagrange's Interpolation discussed in Manembu et al. (2015). Other regression techniques like spline take only a portion of the X value which is not suited for this work. Similarly regression techniques based on logarithm work well when the relationship between X and Y are linear. Since this is not the case in this work, the experiments have been conducted with Lagrange's Interpolation and it resulted in good results.

**Figure 1** Architecture of the proposed model (see online version for colours)



The basic architecture of the proposed model is shown in Figure 1. After the sequential execution of the earlier phases in which unstructuredness and missing data are handled, the input is given for prediction.

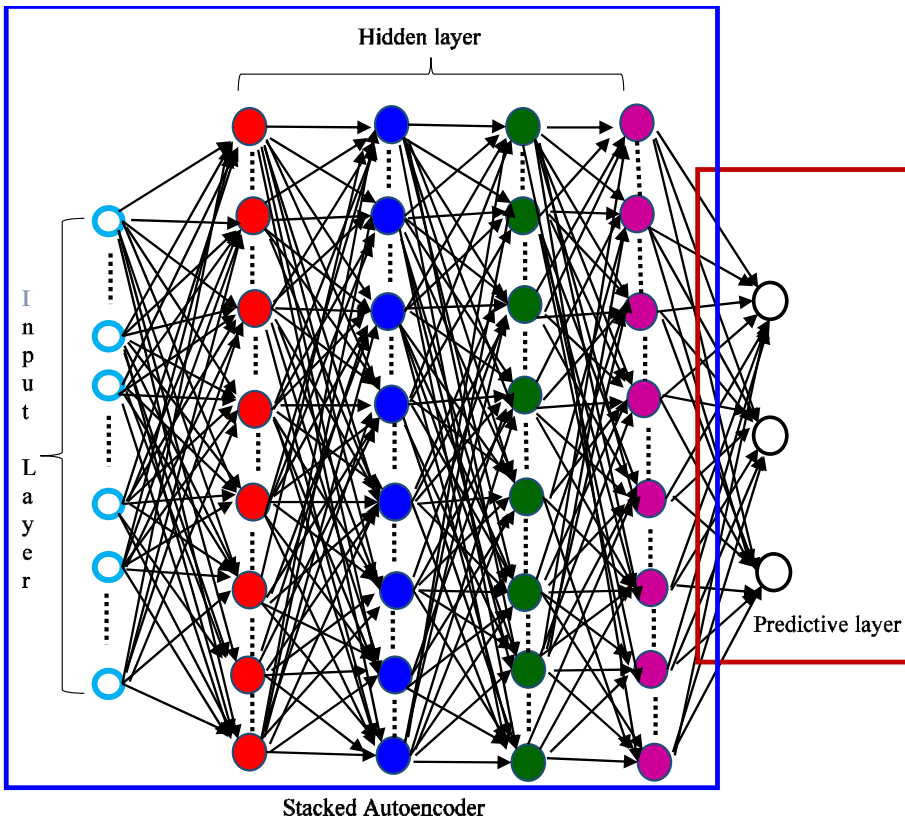
### 2.3 Predicting the progression of ALS

The processed data is next subjected to a predictive analysis that is designed using deep learning approach. The prime motivation behind adoption of deep learning technique is to achieve cost effective computation with higher gain in the performance factors in terms of accuracy. The deep neural network (DNN) based optimisation techniques were also found to exhibit fast convergence speed while taking very less computation time for training and classification. A DNN is composed of three different layers which includes,

- 1 input layer
- 2 hidden layers
- 3 output layer.

There can be many hidden layers as in Figure 2.

**Figure 2** Deep architecture for prediction of ALS (see online version for colours)



The implementation of the proposed system model has been carried out in a cloud based environment with shared cluster. The parameters considered in the study are age, BP, sugar, and ALSFRS. The number of true positives and true negatives obtained were also high when compared with traditional methods. It is assumed that healthcare facility already uses cloud server in order to stack out the generated information.

The model also harnesses the potential of optimisation technique using deep learning approach that further offers cost effective training operation to increase the accuracy of the proposed predictive model. The outcome of the study shows a considerable improvement in prediction accuracy.

The prime reason behind adoption of DNN in this proposed study is that DNN does not pose higher dependency on larger training datasets during the training phase. Hence, it is found to be most suitable operation in our case of medical data processing and analysis.

#### 2.4 Algorithm for predictive analysis

---

Input: inputData, hiddensize1, hiddensize2, lambda, beta, sparsityparam

Output: Classified Data

Step 1 Start

Step 2 Train the First layer on raw inputs

Step 3  $\theta_1 \leftarrow \text{initializeparameters}(\text{hiddensize1}, \text{inputsize})$

Step 4  $\text{autoencode1} \leftarrow \text{trainautoencoder}(\text{inputsize}, \text{hiddensize1}, \lambda, \beta, \text{traindata}, \theta_1)$

Step 5  $\text{feature1} \leftarrow \text{encode}(\text{autoencode1}, \text{hiddenL1}, \text{inputsize}, \text{inputData})$

Step 6 Initialize the second autoencoder

Step 7  $\theta_2 \leftarrow \text{initializeparameters}(\text{hiddensize2}, \text{hiddensize1})$

Step 8  $\text{autoencode2} \leftarrow \text{trainautoencoder}(\text{hiddensize1}, \text{hiddensize2}, \lambda, \beta, \text{sparsityparam}, \text{feature1}, \theta_2)$

Step 9  $\text{feature2} \leftarrow \text{encode}(\text{autoencode2}, \text{features})$

Step 10 The output of the final layer becomes the input to the Softmax Classifier

Step 11 Predict the progression of the disease.

Step 12 End

---

Algorithm 2.1 takes the input of finally processed data, hidden layer size, sparsity parameter, and weight decay parameter (Line-1). The advantage of using Deep Learning architecture is that the entire process like feature extraction, adjusting of weights is automated. The algorithms make use of stacked autoencoders in order to obtain  $\theta_1$  (Line-2). A stacked autoencoder is a network which consists of a number of neurons where the outputs of one layer become the inputs to the next layer. When there are multiple layers of such networks stacked on top of one another, it becomes a stacked autoencoder. Let  $x(y)$  be the output of the deepest layer of hidden units. The features obtained from  $x(y)$  can be used for classification purpose by inputting to a softmax classifier. A random initialisation of the parameters is carried out on the basis of hidden size and input size (Line-3).

The stacked auto encoder is a deep learning model. It will use a layer of auto encoders to build a deep network. In deep nets we represent everything in a vector format. Thus  $\theta$  is obtained as a single dimensional vector, the size of which depends on



The value from 101 to 115 all zeros because hidden size is 10 and visible size is 5

- 2 `autoencode1<-trainautoencoder(inputsize, hiddensize1, lambda,beta,traindata, theta1)`  
Autoencode is a 115\*1 vector of values
- 3 `feature1<-encode(autoencode1,hiddenL1, inputsize, inputData)`

**Table 3** Features generated from input vector

2) <code>autoencode1&lt;-trainautoencoder(inputsize, hiddensize1, lambda,beta,traindata, theta1)</code> Autoencode is a 115*1 vector of values		3) <code>feature1&lt;-encode(autoencode1,hiddenL1, inputsize, inputData)</code>	
-0.2637	Similarly the values are	3.0638e-74	4.8416e-61
0.4324	zeros from 100 to 115	1	1
-0.0537		1.4562e-21	2.6133e-15
-0.3648		8.6141e-19	1.011e-12
0.4091		-----	-----
-0.6096		-----	
-----		1	1
0		1	1
0		0.0334	0.9995
0		0.6386	0.9354
0			
0			
0			
0			
0			

Feature1 shown in Table 3 is a 10\*2 array. 10 is number of hidden layers and 2 inputs are considered. The features from the stacked auto encoder are used here for prediction by feeding  $g(x_i)$  to the softmax classifier. The number of hidden layers used in this network is ten. The weights and biases are passed as raw input to the first layer to obtain a vector  $\theta_1$  in line 1. This is passed through the autoencoder to obtain certain features shown in lines 3 and 4. The second auto encoder is also trained on similar grounds but here the output of the first encoder is also given as its input parameter.

- 5 Initialise the second autoencoder
- 6 `Theta2<-initializeparameters(hiddensize2, hiddensize1)`. This is a 120\*1 vector because hidden size  $l_1 = 10$  size  $l_2 = 10$
- 7 `Autoencode2<- trainautoencoder(hiddensize1, hiddensize2, lambda, sparsityparam, beta, feature1, theta2)`.

**Table 4** Features of the second autoencoder being computed

6) Theta2<- initializeparameters(hiddensize2, hiddensize1). This is a 120*1 vector because hidden size l1 = 10 size l2 = 10	7) Autoencode2<- trainautoencoder(hiddensize1, hiddensize2, lambda, sparsityparam, beta, feature1, theta2)
-0.140	-3.6915e-05
0.1082	2.8564e-04
0.0614	1.6216e-04
0.3987	0.0011
-----	6.8352e-04
0	-----
0	5.3863
0	5.2002
0	1.2563
0	1.5725
0	

8 Feature2<-encode(autoencode2,feature1)

**Table 5** Mapping the output into a 10\*2 vector

0.3590	0.1275
0.3347	0.1111
0.3487	0.1179
0.3644	0.1899
0.3432	0.1161
0.3641	0.1130
0.3195	0.3791
0.3591	0.1281
0.3578	0.1322
0.3492	0.1840

9 ofsoftmaxmodel<-softmaxtrain(hiddenL2, Noc,lambda,feature2,labels)

It's a 2\*10 Vector

**Table 6** Output of Softmax classifier

-0.0083	-0.00035	0.0016	-0.0072	0.0020	0.0059	0.0065	0.0061	-0.0030	0.0010
0.0028	-0.0033	-0.0022	-0.0053	0.0003	-0.0017	0.0021	0.00039	-0.0076	0.0039

After summation

**Table 7** Predicted output

0.00407	-0.01061
---------	----------



10 Pred

**Table 8** Predicted output rounded

1	0
---	---

The prediction now classifies as class 1 or 0 depending on the summation value. If summation is positive it belongs to class 1 (fast progressing) else it belongs to class 0 (slow progressing).

### 3 Experimental results and discussion

#### 3.1 Performance parameters

The performance parameters to evaluate the efficiency of this algorithm are true positive, true negative, false positive, false negative, sensitivity, specificity and accuracy. True positive is the number of ALS patients progressing fast towards the disease. True negative is the number of Non ALS patients progressing slowly towards the disease. False positive is the number of progressing slowly towards the disease. False negative is the number of non-ALS patients progressing fast towards the disease.

The analysis is made on the basis of allowing 60% of samples for training procedure and another 40% samples for the testing process. The entire work is analysed on 1,047,253 patient records. Out of this, 629,539 records are used to train the system and 417,714 records are used to test the system. The train data consists of 377,723 ALS patients and 251,816 non-ALS patients. On the same ground, test data consists of 250,628 ALS patients and 167,086 non-ALS patients. The dataset obtained from ALS contains only ALS patients. In order to test for non-ALS patient, 418,902 records of ALS patients are taken and converted to non-ALS by changing the value of ALS-FRS, BP and Sugar parameters.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FP}} * 100 \quad (3)$$

true positive rate or positive predictive rate (TPR) can be defined as the percentage of data tuples that the classifier labelled as positive is actually positive.

False positive rate or negative predictive rate (FPR) can be defined as the percentage of data tuples that the classifier labelled as negative is actually negative.

$$\text{FPR} = \frac{\text{TN}}{\text{TN} + \text{FN}} * 100 \quad (4)$$

Accuracy is the proportion of records correctly classified

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (5)$$

Sensitivity is the fraction of true values correctly identified as present

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

Specificity is the fraction of false values correctly identified as present

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{7}$$

**Figure 3** Abstract representation of the prediction model (see online version for colours)

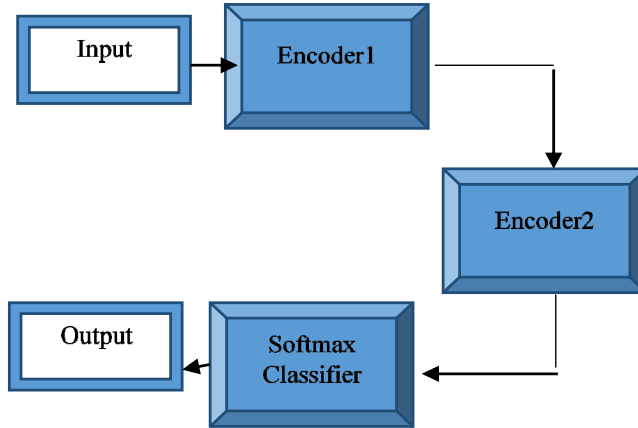


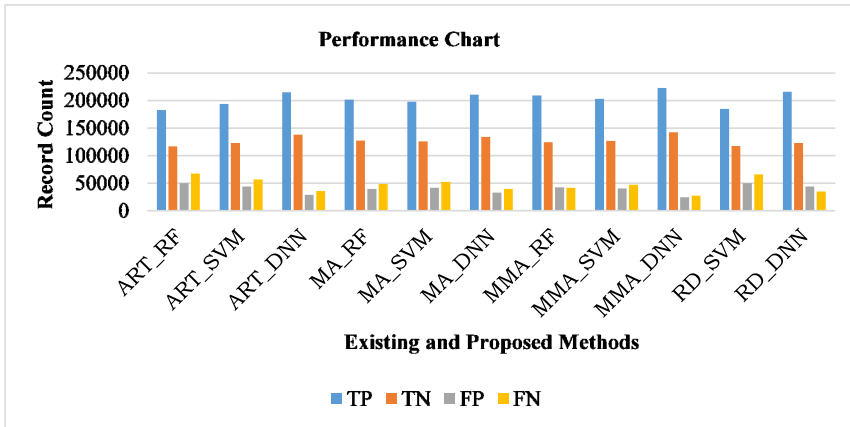
Figure 3 shows the abstract representation of the proposed model. The model consists of two auto encoders. The objective of having two encoders is that better features will be extracted. The task of the encoder is to learn a representation based on its features and the classifier will classify the input as fast progressing or slow progressing based on the learnt features. The softmax classifier is the classifier used with autoencoders for prediction. As the proposed system targets to optimise the performance for large information analytics, it is good enough to be compared with similar often used techniques of optimisation.

**Table 9** Confusion matrix

	<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>
ART_RF	182,958	116,960	50,126	67,670
ART_SVM	193,861	123,180	43,906	56,767
ART_DNN	214,919	138,163	28,923	35,709
MA_RF	201,816	127,281	39,805	48,812
MA_SVM	198,212	125,681	41,405	52,416
MA_DNN	210,912	133,911	33,175	39,716
MMA_RF	209,234	124,382	42,704	41,394
MMA_SVM	203,426	126,698	40,388	47,202
MMA_DNN	223,058	142,304	24,782	27,570
RD_SVM	184,628	117,218	49,868	66,000
RD_DNN	215,783	123,043	44,043	34,845

Table 9 shows the confusion matrix for existing and proposed systems. The graph in Figure 4 shows that proposed prediction process works well if the input is given as cleaned data, i.e., data free from issues related to variety, veracity, and velocity problems.

**Figure 4** Graph of TP, TN, FP, FN for existing and proposed methods (see online version for colours)



The base paper on which this study began predicted the accuracy at 66% using RF algorithm (Ko et al., 2014). For the purpose of result analysis, the study considers implementing the proposed work using RF and observes the outcomes. In Table 9, ART\_RF is the prediction outcome using RF after filling data using auto regression tree (ART) (Sindhu and Hegde, 2015). Similarly, ART\_SVM indicates the prediction outcome of support vector machine (SVM) after filling values through ART and ART\_DNN indicates the prediction accuracy on applying DNNs after filling values through ART. On the same ground, MA\_RF is the accuracy using RF after filling values through modified average. MMA\_RF is the accuracy using RF after filling values through MMA. RD\_DNN is the accuracy of applying DNNs on the raw data without filling any missing values. The various algorithms for filling missing values using ART, moving average (MA) and MMA are shown in Sindhu and Hegde (2015).

This outcome will assist to visualise the comparative analysis of the proposed system using sparse autoencoder and the existing mechanism with respect to accuracy performance as the performance parameters.

True positives are the number of ALS patients progressing fast towards the disease. True negatives are the number of Non-ALS patients progressing slowly towards the disease. False positives are the number of Non-ALS patients wrongly classified as progressing fast towards the disease. False negatives are the number of ALS patients wrongly classified as progressing slowly towards the disease. Table 9 shows that the number of true positives and true negatives for MMA using DNNs (MMA\_DNN) is the highest. The graph in Figure 4 depicts the same.

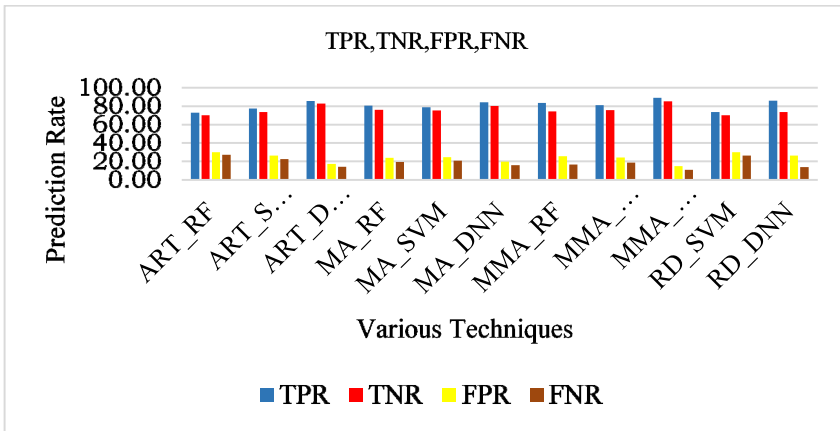
**Table 10** Rate of TP, TN, FP, FN for various methods

	<i>TPR</i>	<i>TNR</i>	<i>FPR</i>	<i>FNR</i>
ART_RF	73.00	70.00	30.00	27.00
ART_SVM	77.35	73.72	26.28	22.65
ART_DNN	85.75	82.69	17.31	14.25
MA_RF	80.52	76.18	23.82	19.48
MA_SVM	79.09	75.22	24.78	20.91
MA_DNN	84.15	80.14	19.86	15.85
MMA_RF	83.48	74.44	25.56	16.52
MMA_SVM	81.17	75.83	24.17	18.83
MMA_DNN	89.00	85.17	14.83	11.00
RD_SVM	73.67	70.15	29.85	26.33
RD_DNN	86.10	73.64	26.36	13.90

The TPR, true negative rate (TNR), FPR and false negative rate (FNR) are shown in Table 10. False Positive is where the model incorrectly predicts the positive class and true negative is when the model correctly predicts the negative class and false negative is when the model incorrectly predicts the negative class. In an efficient model the false negative and false positive should be low. The experiments were conducted on the data obtained from ART, modified average and MMA (Müller et al., 2018). Training was done and then the prediction algorithms were applied to validate the accuracy.

The outcome shows that the proposed mechanism accomplishes better true positive with respect to statistical rating (probability) perspective, while the existing systems offers slightly reduced accuracy in performance compared to proposed system. The prime reason behind this is that the proposed system contains quality data which is obtained by filling in all the missing values and performs a sequential operation where output of first operation becomes input of the second process.

**Figure 5** Graph indicating comparison of TPR, TNR, FPR, FNR for existing and proposed system (see online version for colours)



Graph in Figure 5 depicts the above information in Table 10.

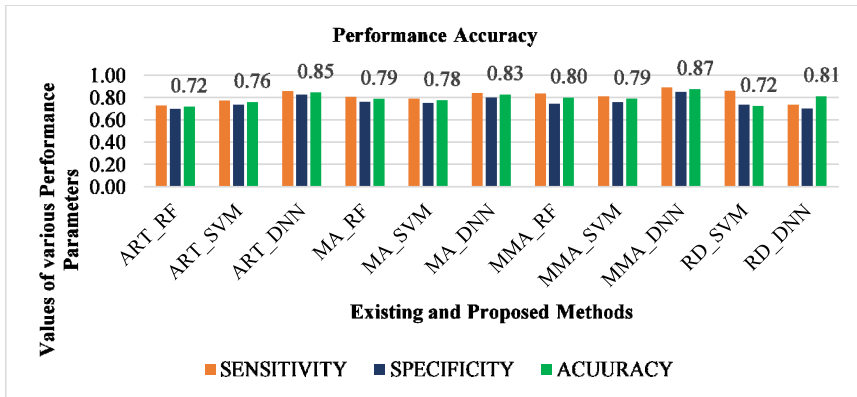
This shows that quality of data plays a very important role in the accuracy of prediction. The sensitivity, specificity and accuracy values for proposed MMA\_DNN and existing methods are also analysed and are given in Table 11. The graph for Table 11 is shown in Figure 6. From the graph, it is evident that highest accuracy of 87% is obtained when using MMA-DNN method. The values for Sensitivity and Specificity are highest for the same method. Table 9 shows the values for SVM and RF when using ART, MA and MMA algorithms.

**Table 11** Sensitivity, specificity and accuracy values for proposed and existing methods

	<i>Sensitivity</i>	<i>Specificity</i>	<i>Accuracy</i>
ART_RF	0.73	0.70	0.72
ART_SVM	0.77	0.74	0.76
ART_DNN	0.86	0.83	0.85
MA_RF	0.81	0.76	0.79
MA_SVM	0.79	0.75	0.78
MA_DNN	0.84	0.80	0.83
MMA_RF	0.83	0.74	0.80
MMA_SVM	0.81	0.76	0.79
<i>MMA_DNN</i>	<i>0.89</i>	<i>0.85</i>	<i>0.87</i>
RD_SVM	0.74	0.70	0.81
RD_DNN	0.86	0.74	0.72

True positives are the number of ALS patients progressing fast towards the disease. True negatives are the number of non-ALS patients progressing slowly towards the disease. False positives are the number of non-ALS patients wrongly classified as progressing fast towards the disease. False negatives are the number of ALS patients wrongly classified as progressing slowly towards the disease. Table 9 shows that the number of true positives and true negatives for MMA\_DNN is the highest. The graph in Figure 5 depicts the same.

**Figure 6** Prediction accuracy performance (see online version for colours)



The outcome of the proposed system shows faster convergence whilst making transition from one algorithm to another until prediction. From the graph it is evident that the proposed system offers 87% accuracy which is the highest in comparison with the traditional systems. In addition to this, there is a very high increase in prediction accuracy when compared to the base paper which delivered 66% (Ko et al., 2014). When DNNs was applied on raw data, the accuracy was 81% which is far less than what was achieved using cleaned data. Therefore, the proposed system offers better accuracy over traditional techniques to prove its cost effectiveness.

**Complexity of the algorithm:** We assume the input-vector can be described as:  $x \in R_n$  where the first element is the bias unit:  $x_0 = 1$ . The input is treated in the same as any other activation matrix, and has the index:  $x = a(0)$ . The zeroth element,  $a(0)$  is as usual the bias unit with a value of 1.

The output at each node can be written as

$$z(k) = \theta(k) a(k-1)$$

$$z(k) = R_1 \times (m_1 \theta(k) \in R_{m \times n})$$

$$a(k) = g(z(k))$$

where  $g(x)$  is the activation function which is evaluated elementwise. Therefore  $a(k)$  has the same dimensions as  $z(k)$

For each layer a matrix multiplication, and an activation function is computed. A naive matrix multiplication has a asymptotic run-time of  $O(n^3)$ , and since  $g(x)$  is an elementwise function, its run-time is  $O(n)$ . By analysing the dimensions of a this neural network, it was found that

$$\theta(0) \in R_n(0) \times 1$$

$$\theta(1) \in R_n(0) \times n(0)$$

$$\theta(2) \in R_n(2) \times n(1)$$

More generally, we can write:

$$\theta(k) = \begin{cases} R_n(k) \times 1 & \text{if } k = 0 \\ R_n(k) \times n(k-1) & \text{if } k > 0 \end{cases}$$

where  $n(k)$  is the number of neurons including the bias unit in layer  $k$ .

We can write input, hidden and output layer as

$$a = n(k)$$

$$z = n(k-1)$$

$$d = n(k-2)$$

From this we find that:

$$n_{mul} = \sum_{k=2}^{n_{layers}} n(k)n(k-1)n(k-2) + (n(1) + n(0)l)$$

$$n_g = \sum_{k=2}^{n_{layers}} n(k)$$

where  $n^{mul}$  is the number of multiplications performed, and  $n^g$  is how many times we apply the activation function. We have assumed that  $a(k)$  has the same dimension as  $\theta(k)$  (Küffner et al., 2015).

This gives

$$time = n^{mul} + n^g$$

$$time = \sum_n^{layers} (n(k)n(k-1)n(k-2)) + (n(1)n(0)n(0)l)$$

$$+ \sum_n^{layers} n(k)$$

When analysing matrix algorithms, it's common to assume that the matrices are quadratic; that is they have the same number of rows, as columns. By doing this, we get

$$n^{mul} = n^{layers} \cdot n^3$$

If we once again assume that there are the same number of neurons in each layer, and that the number of layers equal the number of neurons in each layer we find:

$$n_{mul} = O(n \cdot n^3) = O(n^4)$$

The same can be done for the activations:

$$n_g = n_{layers} \cdot n = O(n^2)$$

The total run-time therefore becomes:

$$O(n^4 + n^2) = O(n^4)$$

Because for all  $n > 1 | n^4 + n^2 \leq 2n^4$ .

## 4 Conclusions

It is evident from this work that if the quality of data is increased, the accuracy levels also increases. Improvement in quality is obtained by addressing the issue of missing values or discontinuity in data. Data has been filled using auto regression tree, moving average and MMA. It has been proven in the earlier studies that MMA produces good results. Prediction using deep nets has proved to deliver the highest accuracy. Complexity of the algorithm was also found to be  $O(n^4)$ . The findings showed that accuracy prediction of ALS on raw data was 66% whereas that on cleaned and complete data was 87% when used with deep learning algorithms.

## References

- Achiepo, O.Y.M., N'Guessan, B.G. and Brou, K.M. (2015) 'Similarity measure in the case-based reasoning systems for medical diagnostics in traditional medicine', *International Journal of Computer Science Issues (IJCSI)*, Vol. 12, No. 2, p.239.
- Ko, K.D., El-Ghazawi, T., Kim, D. and Morizono, H. (2014) 'Predicting the severity of motor neuron disease progression using electronic health record data with a cloud computing big data approach', in *2014 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*, IEEE, May, pp.1–6.
- Küffner, R., Zach, N., Norel, R., Hawe, J., Schoenfeld, D., Wang, L. and Cudkowicz, M. (2015) 'Crowdsourced analysis of clinical trial data to predict amyotrophic lateral sclerosis progression', *Nature Biotechnology*, Vol. 33, No. 1, p.51.
- Manembu, P., Kewo, A. and Welang, B. (2015) 'Missing data solution of electricity consumption based on Lagrange interpolation case study: IntelligEnSia data monitoring', in *2015 International Conference on Electrical Engineering and Informatics*, ICEEI, August, pp.511–516.
- Marwala, T. (2009) 'Estimation of missing data using neural networks and decision trees' in *Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques*, IGI Global, pp.233–255.
- Matsumura, S. (1992) U.S. Patent No. 5,148,499, US Patent and Trademark Office, Washington, DC.
- Meijering, E.H., Niessen, W.J. and Viergever, M.A. (2001) 'Quantitative evaluation of convolution-based methods for medical image interpolation', *Medical Image Analysis*, Vol. 5, No. 2, pp.111–126.
- Menon, S.P. (2019) 'Applying lagrange model to fill data during big data streaming', in *International Conference on Sustainable Communication Networks and Application*, Springer, Cham, July, pp.97–107.
- Menon, S.P. and Hegde, N.P. (2015) 'A survey of tools and applications in big data', in *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, IEEE, January, pp.1–7.
- Müller, H.P., Agosta, F., Riva, N., Spinelli, E.G., Comi, G., Ludolph, A.C. and Kassubek, J. (2018) 'Fast progressive lower motor neuron disease is an ALS variant: a two-centre tract of interest-based MRI data analysis', *NeuroImage: Clinical*, Vol. 17, pp.145–152.
- Ong, M.L., Tan, P.F. and Holbrook, J.D. (2017) 'Predicting functional decline and survival in amyotrophic lateral sclerosis', *PloS one*, Vol. 12, No. 4, pp.1–16.
- PRO\_ACT Database [online] <https://nctu.partners.org/ProACT> (accessed 12 August 2019).
- Schuster, C., Hardiman, O. and Bede, P. (2017) 'Survival prediction in amyotrophic lateral sclerosis based on MRI measures and clinical characteristics', *BMC Neurology*, Vol. 17, No. 1, p.73.
- Sindhu, C.S. and Hegde, N.P. (2015) 'A framework to handle data heterogeneity contextual to medical big data', in *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, IEEE, December, pp.1–7.
- Sindhu, C.S. and Hegde, N.P. (2017) 'A novel integrated framework to ensure better data quality in big data analytics over cloud environment', *International Journal of Electrical and Computer Engineering*, Vol. 7, No. 5, pp.2798–2805.